Chapter 11
Policer Configuration

To configure policers, you include statements at the [edit firewall] hierarchy level of the configuration:

```
[edit]
firewall {
    policer policer-name {
        filter-specific;
        if-exceeding {
            bandwidth-limit bps;
            bandwidth-percent number;
            burst-size-limit bytes;
        }
        then {
            policer-action;
        }
    }
    family family-name {
        filter filter-name {
            accounting-profile name;
            interface-specific;
        }
        prefix-action name {
            count;
            destination-prefix-length prefix-length;
            policer policer-name;
            source-prefix-length prefix-length;
            subnet-prefix-length prefix-length;
        }
        prefix-policer {
            policer policer-name;
        }
    }
}
```

The following sections explain the tasks required for configuring policers and provide configuration examples:

Minimum Policer Configuration on page 186

Configure Policers on page 187

Apply an Interface Policer on page 195

Examples: Configure Policing on page 196

Minimum Policer Configuration

To configure a policer, you must perform at least the following tasks:

Configure policers—To configure policers, include the policer statement at the [edit firewall] hierarchy level. After policers are defined, you reference them in the then clause of a term:

```
[edit]
firewall {
    policer policer-name {
        filter-specific;
        if-exceeding {
            bandwidth-limit bps;
            bandwidth-percent number;
            burst-size-limit bytes;
        }
        then {
            policer-action;
        }
    }
    family family-name {
        filter filter-name  {
    }
}
```

Add actions, such as accept, discard, or next term, or action modifiers, such as count or log.

Apply the policers to an interface to activate them.

The policer is applied to the packet first, and if the packet exceeds the defined limits, the actions of the then clause of the policer are applied. If the result of the policing action is not a discard, the remaining components of the then clause of the term are applied.

To display statistics about a filter statement policer configuration, use the show policers command.

Configure Policers

To configure term-specific policers, include the policer statement:

```
policer policer-name {
    if-exceeding {
        bandwidth-limit rate;
        bandwidth-percent number;
        burst-size-limit bytes;
    }
    then {
        policer-action;
    }
}
```

You can configure the policer statement at the [edit firewall] hierarchy level. The following sections describe the components of the policer statement and provide policer configuration examples:

Configure Rate Limiting on page 187

Configure a Policer Action on page 188

Configure Multifield Classification and Policing on page 189

Configure Filter-Specific Policers on page 189

Configure Prefix-Specific Action on page 190

Examples: Classify Traffic on page 194

Configure Rate Limiting

To specify the rate limiting part of a policer, include an if-exceeding statement at the [edit firewall policer *policer-name*] hierarchy level:

```
[edit firewall policer policer-name]
if-exceeding {
    bandwidth-limit bps;
    bandwidth-percent number;
    burst-size-limit bytes;
}
```

You specify the bandwidth limit in bits per second. You can specify the value as a complete decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000). There is no absolute minimum value for bandwidth limit, but any value below 61,040 bps will result in an effective rate of 30,520 bps. The maximum bandwidth limit is 4.29 Gbps.

You can rate-limit based upon port speed. This port speed can be specified by a bandwidth percentage in a policer. You must specify the percentage as a complete decimal number between 1 and 100.

You cannot rate-limit based on bandwidth percentage for aggregate, tunnel, and software interfaces. The bandwidth percentage policer cannot be used for forwarding table filters. This can only be used for interface specific filters.

The maximum burst size controls the amount of traffic bursting allowed. To determine the value for the burst-size limit, the preferred method is to multiply the bandwidth of the interface on which you are applying the filter by the amount of time you allow a burst of traffic at that bandwidth to occur; for example, 5 milliseconds.

burst size = bandwidth x allowable time for burst traffic

If you do not know the interface bandwidth, you can multiply the maximum transmission unit (MTU) of the traffic on the interface by 10 to obtain a value. For example, the burst size for an MTU of 4700 would be 47,000 bytes. At minimum, burst size should be at least 10 interface MTUs. The maximum value for the burst-size limit is 100 MB.

For a sample filter configuration for rate limiting, see “Examples: Configure Policing” on page 196.

Configure a Policer Action

If a packet does not exceed its rate limits, it is processed further without being affected. If the packet exceeds its limits, it is handled in one of two ways, depending on what you specify:

Discarded

Marked for subsequent processing based on its loss priority and forwarding class

To configure a policer action, include the then statement:

```
then {
    policer-action;
}
```

You can configure policer actions at the [edit firewall policer *policer-name*] hierarchy level. Policer actions include the following:

discard—Discard a packet that exceeds the rate limits.

loss-priority—Set the loss priority level to low or high.

forwarding-class—Specify the forwarding class to any class name already configured for the forwarding class.

Example: Configure a Policer Action

Discard any packet that exceeds a bandwidth of 300 Kbps and a burst-size limit of 500 KB:

```
[edit firewall]
policer p1 {
    if-exceeding {
        bandwidth-limit 300k;
        burst-size-limit 500k;
    }
    then {
        discard;
    }
}
```

Configure Multifield Classification and Policing

You can configure *multifield classifier s* within a firewall filter to set the packet's forwarding class and packet loss priority. You can also apply policers to packets matching some classification term. The policing action might affect the resulting forwarding class, packet loss priority, and accept or drop status. For more information, see the *JUNOS Internet Softw are Configur ation Guide: Interf aces and Class of Service* .

To configure the forwarding class and loss priority, include the forwarding-class *class-name* and loss-priority *level* actions at the [edit firewall filter *filter-name* term *term-name* then] or [edit firewall filter *filter-name* policer *policer-name* then] hierarchy level:

```
then {
    loss-priority level;
    forwarding-class class-name;
}
```

You can specify one or both of the following actions:

loss-priority—Set the loss priority level to low or high.

forwarding-class—Specify the forwarding class to any class name already configured for the forwarding class.

For more information about forwarding class and loss priority, see the *JUNOS Internet Software Configur ation Guide: Interf aces and Class of Service* .

Configure Filter-Specific Policers

You can configure filter-specific policers within the firewall configuration. Filter-specific policers allows you to configure policers and counters for a specific filter name.

To configure filter-specific policers, include the filter-specific statement at the [edit firewall policer *policer-name*] hierarchy level:

```
filter-specific;
```

If the filter-specific statement is not configured, then the policer defaults to a term-specific policer.

You can apply the filter-specific policers to the family inet.

To apply filter-specific policers, include the prefix-policer statement at the [edit firewall family inet] hierarchy level:

```
prefix-policer {
    policer policer-name;
}
```

When you configure this statement, the policer referenced is the filter-specific policer.

Configure Prefix-Specific Action

You can configure prefix-specific action within the firewall configuration. Prefix-specific action allows you to configure policers and counters for specific addresses or ranges of addresses. This allows you to essentially create policers and counters on a per-prefix level.

To configure prefix-specific actions, include the count, destination-prefix-length *prefix-length*, policer *policer-name*, source-prefix-length *prefix-length*, and subnet-prefix-length *prefix-length* statements at the [edit firewall family inet prefix-action *name*] hierarchy level:

```
prefix-action name {
    count;
    destination-prefix-length prefix-length;
    policer policer-name;
    source-prefix-length prefix-length;
    subnet-prefix-length prefix-length;
}
```

Prefix-specific action is supported for IPv4 inet address family.

The following section describes the statement options:

count—Specify this option to enable a prefix-specific counter.

destination-prefix-length—Refer to the destination address range specified for a prefix-specific policer or counter.

policer—Specify this option to enable a set of prefix-specific policers.

source-prefix-length—Refer to the source address range specified for a prefix-specific policer or counter.

subnet-prefix-length—Refers to the total address range of the subnet supported.

The source or destination prefix length must be larger than the subnet prefix length.

Examples: Configure Prefix-Specific Actions

The following example creates a prefix-specific policer operating on the source address and applies it to the input interface.

```
[edit]
firewall {
    policer host-policer {
        filter-specific;
        if-exceeding {
            bandwidth-limit bps;
            burst-size-limit bytes;
        }
        then {
            discard;
        }
    }
    family inet {
        prefix-action ftp-policer-set {
            count;
            destination-prefix-length 32;
            policer host-policer;
            subnet-prefix-length 24;
        }
        filter filter-ftp {
            term term1{
                from {
                    destination-address 10.10.10/24;
                    destination-port ftp;
                }
                then {
                    prefix-action ftp-policer-set;
                }
            }
        }
    }
}
```

The following example filters all packets going to the /24 subnet, letting it pass to the prefix-specific action policers. In the policer set, the last octet of the source address field of the packet is used to index into the respective prefix-specific action policers.

```
[edit]
firewall {
policer 1Mbps-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 63k;
        }
    }
    family inet {
        prefix-action per-source-policer {
            policer 1Mbps-policer;
            subnet-prefix-length 24;
            source-prefix-length 32;
        }
    }
```

```
    filter limit-all-hosts {
        term one {
            from {
                source-address {
                    10.10.10.0/24;
                }
            }
            then prefix-action per-source-policer;
        }
    }
}
```

The following example filters all packets, letting it pass to the prefix-specific action policers. In the policer set, the last octet of source address field of the packet is used to index into the corresponding prefix-specific action policers.

```
[edit]
firewall {
    policer 1Mbps-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 63k;
        }
    }
    family inet {
        prefix-action per-source-policer {
            policer 1Mbps-policer;
            subnet-prefix-length 24;
            source-prefix-length 32;
        }
    }
    filter limit-all-hosts {
        term one {
            then prefix-action per-source-policer;
        }
    }
}
```

In the following example, packets belonging to the 10.10.10.0/24 subnet are subjected to policing by the prefix-specific action policers.There are 128 policers defined in the policer set. Therefore the /24 subnet can be thought of being split into two /25s subnets, both of them sharing the same prefix-specific action set.

```
[edit]
firewall {
    policer 1Mbps-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 63k;
        }
    }
    family inet {
        prefix-action per-source-policer {
            policer 1Mbps-policer;
            subnet-prefix-length 25;
            source-prefix-length 32;
        }
    }
```

```
    filter limit-all-hosts {
        term one {
            from {
                source-address {
                    10.10.10.0/24;
                }
            }
            then prefix-action per-source-policer;
        }
    }
}
```

The following example defines 256 policers based on the last octet of the source address field. However, you are only allowing a subset of that to pass through the match condition. As a result, only the lower half of the set is used.

```
[edit]
firewall {
    policer 1Mbps-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 63k;
        }
    }
    family inet {
        prefix-action per-source-policer {
            policer 1Mbps-policer;
            subnet-prefix-length 24;
            source-prefix-length 32;
        }
    }
    filter limit-all-hosts {
        term one {
            from {
                source-address {
                    10.10.10.0/25;
                }
            }
            then prefix-action per-source-policer;
        }
    }
}
```

The following example accepts packets from 10.10.10/24 and 11.11/16 subnets and subject them to policing by the same set of prefix-specific action policers. The policers are shared by packets across both subnets. There is a one to one correspondence between the 10.10.10/24 subnet. For the 11.11/16 subnet, there is a many to one correspondence. Here, each of the 11.11.0/24, 11.11.1/24, 11.11.2/24 ...11.11.255/24 share the same prefix-specific action set.

```
[edit]
firewall {
    policer 1Mbps-policer {
        if-exceeding {
            bandwidth-limit 1m;
            burst-size-limit 63k;
        }
    }
```

```
    family inet {
        prefix-action per-source-policer {
            policer 1Mbps-policer;
            subnet-prefix-length 24;
            source-prefix-length 32;
        }
    }
    filter limit-all-hosts {
        term one {
            from {
                source-address {
                    10.10.10/24;
                    11.11/16;
                }
            }
            then prefix-action per-source-policer;
        }
    }
}
```

Examples: Classify Traffic

Classify expedited forwarding traffic:

```
[edit]
firewall {
    policer ef-policer {
        if-exceeding {
            bandwidth-limit 300k;
            burst-size-limit 50k;
        }
        then {
            discard;
        }
    }
    term ef-multifield {
        then {
            loss-priority low;
            forwarding-class expedited-forwarding;
            policer ef-policer;
        }
    }
}
```

Classify assured forwarding traffic:

```
firewall {
    policer af-policer {
        if-exceeding {
            bandwidth-limit 300k;
            burst-size-limit 500k;
        }
        then {
            loss-priority high;
        }
    }
```

```
    term af-multifield {
        then {
            loss-priority low;
            forwarding-class assured-forwarding;
            policer af-policer;
        }
    }
}
```

Apply an Interface Policer

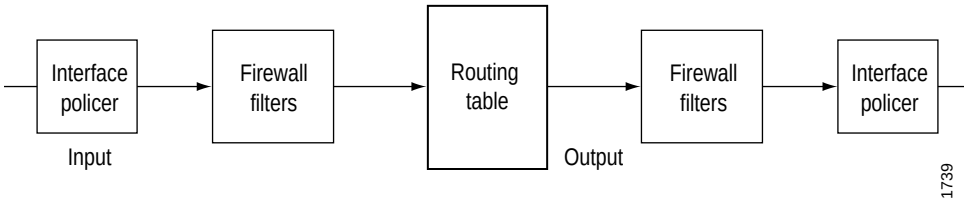
In addition to including policers in firewall filters, you can apply an interface policer that is not part of a firewall filter configuration. An interface policer can be applied to each family on an interface.

To apply an interface policer, include the policer statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family-name*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family-name]
policer {
    input policer-name;
    output policer-name;
}
```

You must first configure the policer at the [edit firewall] hierarchy level before you can apply it to an interface. Both input and output policers are allowed, and can be used in conjunction with existing firewall filters. Input interface policers are evaluated before any input firewall filters. Likewise, output interface policers are evaluated after any output firewall filters (see Figure 12.)

Figure 12: Incoming and Outgoing Interface Policers



To display a policer on a particular interface, issue the show interfaces policers command at the command-line interface (CLI).

Example: Apply an Interface Policer

Apply a policer on circuit cross-connect (CCC) interfaces:

```
[edit interfaces]
so-0/0/0 {
    encapsulation ppp-ccc;
    unit 0 {
        family ccc {
            policer {
                input dragnet;
            }
        }
    }
}
```

Examples: Configure Policing

The following example shows a complete filter configuration containing a policer. It limits all FTP traffic from a given source to certain rate limits. Traffic exceeding the limits is discarded, and the remaining traffic is accepted and counted.

```
[edit]
firewall {
    policer policer-1 {
        if-exceeding {
            bandwidth-limit 400k;
            burst-size-limit 100k;
        }
        then {
            discard;
        }
    }
    term tcp-ftp {
        from {
            source-address 1.2.3/24;
            protocol tcp;
            destination-port ftp;
        }
        then {
            policer policer-1;
            accept;
            count count-ftp;
        }
    }
}
```

The following example shows a complete filter configuration containing two policers, and includes the next term action. Policer policer-1 limits all traffic from a given source to certain rate limits, then sets the forwarding class. Policer policer-2 limits all traffic to a second set of rate limits. Traffic exceeding the limits is discarded; the remaining traffic is accepted.

```
[edit]
firewall {
    policer policer-1 {
        if-exceeding {
            bandwidth-limit 10m;
            burst-size-limit 100k;
        }
        then {
            forwarding-class 0;
        }
    }
    policer policer-2 {
        if-exceeding {
            bandwidth-limit 100m;
            burst-size-limit 100k;
        }
        then {
            discard;
        }
    }
    filter f {
        term term-1 {
            then {
                policer policer-1;
                next term;
            }
        }
        term term-2 {
            then {
                policer policer-2;
                accept;
            }
        }
    }
```

The following example limits all FTP traffic from a given source to certain rate limits, but defines the policer outside the filter, thereby creating a template that can be referenced by more than one filter or more than one term within a filter. Traffic exceeding the limits is discarded, and the remaining traffic is accepted and counted.

```
[edit]
firewall {
    policer policer-1 {
        if-exceeding {
            bandwidth-limit 400k;
            burst-size-limit 100k;
        }
        then {
            discard;
        }
    }
    filter limit-ftp {
        term tcp-ftp {
            from {
                source-address 1.2.3/24;
                protocol tcp;
                destination-port ftp;
            }
            then {
                policer policer-1;
                accept;
                count count-ftp;
            }
        }
    }
}
```

The following example shows a filter intended to thwart denial-of-service (DoS) SYN attacks:

```
[edit]
firewall {
    policer syn-recvd {
        if-exceeding {
            bandwidth-limit 40k;
            burst-size-limit 15000;
        }
        then discard;
    }
    term allow-syn {
        from {
            source-address {
                168.17.12.50/32;            # trusted addresses
            }
        }
        then {
            log;
            accept;
        }
    }
    term limit-syn {
        from {
            protocol tcp;
            tcp-initial;
        }
        then {
            count limit-syn;
            policer syn-recvd;
            accept;
        }
    }
    term default {
        then accept;
    }
}

[edit]                                  # apply filter to lo0 to control traffic to the Routing Engine
interfaces {
    lo0 {
        unit 0 {
            family inet {
                filter {
                input syn-attack;
                }
            }
            address 168.164.4.53/32;
        }
    }
}
```

The following example uses one filter to do the following:

Stop all UDP and ICMP traffic destined to these addresses (in term a).

Send ICMP through the policer (in term b).

Accept ICMP traffic within contract and all other traffic (in term c).

It is important to keep the terms in order; once a packet has a match within the firewall filter, it is not examined in subsequent terms. For example, if you configured the filter to send ICMP traffic through the policer before discarding ICMP and UDP traffic to those addresses, it would not work.

```
[edit firewall]
policer policer-1 {
    if-exceeding {
        bandwidth-limit 200k;
        burst-size-limit 3k;
    }
    then {
        loss-priority 1;
        forwarding-class 1;
    }
}
term a {
    from {
        destination-address {
            104.126.50.2/23;
            188.130.12.1/23;
            163.82.16.0/24 except;
            163.82.0.3/18;
        }
        protocol [icmp udp];
    }
    then {
        count packets-dropped;
        discard;
    }
}
term b {
    from {
        protocol icmp;
    }
    then policer policer-1;
}
term c {
    then accept;
}
```